(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(54) Title: COMPUTERIZED GAMING SYSTEM, METHOD AND APPARATUS

(57) Abstract: The present invention in various embodiments provides a computerized wagering game method and apparatus that
features an operating system kernel (201), a system handler application (300) that loads and executes gaming program shared objects
(203) and features nonvolatile storage that facilitates sharing of information between gaming program objects (203). The system
handle (300)further provides an application program interface (206) library of functions callable from the gaming program objects
(203), and facilitates the use of callback functions on change of data stored in nonvolatile memory (204). The nonvolatile storage
(204) also provides protection against loss of game state during a power loss. The system handler application (300) includes a
plurality of device handlers (210) providing an interface to selected hardware and the ability to monitor hardware events.

COMPUTERIZED GAMING SYSTEM, METHOD AND APPARATUS

5   ### Field of the Invention

The invention relates generally to computerized gaming systems, and more specifically to a game code and operating system method and apparatus for use within computerized gaming systems.

10   ### Notice of Co-pending Applications

This application is related to co-pending application 09/405,921 filed September 24, 1999, and to co-pending application 09/520,405, filed March 8, 2000, which are hereby incorporated by reference.

15   ### Background of the Invention

Games of chance have been enjoyed by people for thousands of years and have enjoyed increased and widespread popularity in recent times. As with most forms of entertainment, players enjoy playing a wide variety of games and new games. Playing new games adds to the excitement of "gaming." As is well known in the art and as used
20   herein, the term "gaming" and "gaming devices" are used to indicate that some form of wagering is involved, and that players must make wagers of value, whether actual currency or some equivalent of value, e.g., token or credit.

One popular game of chance is the slot machine. Conventionally, a slot machine is configured for a player to wager something of value, e.g., currency, house token,
25   established credit or other representation of currency or credit. After the wager has been made, the player activates the slot machine to cause a random event to occur. The player wagers that particular random events will occur that will return value to the player. A standard device causes a plurality of reels to spin and ultimately stop, displaying a

1

random combination of some form of indicia, for example, numbers or symbols. If this display contains one of a pre-selected plurality of winning combinations, the machine releases money into a payout chute or increments a credit meter by the amount won by the player. For example, if a player initially wagered two coins of a specific denomination

5   and that player achieved a payout, that player may receive the same number or multiples of the wager amount in coins of the same denomination as wagered.

There are many different formats for generating the random display of events that can occur to determine payouts in wagering devices. The standard or original format was the use of three reels with symbols distributed over the face of the wheel. When the three

10 reels were spun, they would eventually each stop in turn, displaying a combination of three symbols (e.g., with three wheels and the use of a single payout line as a row in the middle of the area where the symbols are displayed). By appropriately distributing and varying the symbols on each of the reels, the random occurrence of predetermined winning combinations can be provided in mathematically predetermined probabilities.

15 By clearly providing for specific probabilities for each of the pre-selected winning outcomes, precise odds that would control the amount of the payout for any particular combination and the percentage return on wagers for the house could be readily controlled.

Other formats of gaming apparatus that have developed in a progression from the

20 pure slot machine with three reels have dramatically increased with the development of video gaming apparatus. Rather than have only mechanical elements such as wheels or reels that turn and stop to randomly display symbols, video gaming apparatus and the rapidly increasing sophistication in hardware and software have enabled an explosion of new and exciting gaming apparatus. The earlier video apparatus merely imitated or

25 simulated the mechanical slot games in the belief that players would want to play only the same games. Early video games therefore were simulated slot machines. The use of video gaming apparatus to play new games such as draw poker and Keno broke the ground for the realization that there were many untapped formats for gaming apparatus.

2

Now casinos may have hundreds of different types of gaming apparatus with an equal number of significant differences in play. The apparatus may vary from traditional three reel slot machines with a single payout line, video simulations of three reel video slot machines, to five reel, five column simulated slot machines with a choice of twenty or

5   more distinct paylines, including randomly placed lines, scatter pays, or single image payouts. In addition to the variation in formats for the play of games, bonus plays, bonus awards, and progressive jackpots have been introduced with great success. The bonuses may be associated with the play of games that are quite distinct from the play of the original game, such as the video display of a horse race with "bets" on the individual

10 horses randomly assigned to players that qualify for a bonus, the spinning of a random wheel with fixed amounts of a bonus payout on the wheel (or simulation thereof), or attempting to select a random card that is of higher value than a card exposed on behalf of a virtual "dealer."

Examples of such gaming apparatus with a distinct bonus feature includes U.S.

15 Patent Nos. 5,823,874; 5,848,932; 5,836,041; U.K. Patent Nos. 2 201 821 A; 2 202 984 A; and 2 072 395A; and German Patent DE 40 14 477 A1. Each of these patents differ in fairly subtle ways as to the manner in which the bonus round is played. British patent 2 201 821 A and DE 37 00 861 A1 describe a gaming apparatus in which after a winning outcome is first achieved in a reel-type gaming segment, a second segment is engaged to

20 determine the amount of money or extra games awarded. The second segment gaming play involves a spinning wheel with awards listed thereon (e.g., the number of coins or number of extra plays) and a spinning arrow that will point to segments of the wheel with the values of the awards thereon. A player will press a stop button and the arrow will point to one of the values. The specification indicates both that there is a level of skill

25 possibly involved in the stopping of the wheel and the arrow(s), and also that an associated computer operates the random selection of the rotatable numbers and determines the results in the additional winning game, which indicates some level of random selection in the second gaming segment.

U.S. Patents Nos. 5,823,874 and 5,848,932 describe a gaming device comprising:
a first, standard gaming unit for displaying a randomly selected combination of indicia,
said displayed indicia selected from the group consisting of reels, indicia of reels, indicia
of playing cards, and combinations thereof;   means for generating at least one signal
5   corresponding to at least one select display of indicia by said first, standard gaming unit;
means for providing at least one discernible indicia of a mechanical bonus indicator, said
discernible indicia indicating at least one of a plurality of possible bonuses, wherein said
providing means is operatively connected to said first, standard gaming unit and becomes
actuatable in response to said signal.   In effect, the second gaming event simulates a
10   mechanical bonus indicator such as a roulette wheel or wheel with a pointing element.

A video terminal is another form of gaming device. Video terminals operate in
the same manner as a conventional slot and video machine, except that a redemption
ticket rather than an immediate payout is dispensed.

The vast array of electronic video gaming apparatus that is commercially available
15   is not standardized within the industry or necessarily even within the commercial line of
apparatus available from a single manufacturer. One of the reasons for this lack of
uniformity or standardization is the fact that the operating systems that have been used to
date in the industry are primitive. As a result, the programmer must often create code for
each and every function performed by each individual apparatus.

20   Attempts have been made to create a universal gaming engine for a gaming
machine and is described in Carlson U.S. patent 5,707,286. This patent describes a
universal gaming engine that segregates the random number generator and transform
algorithms so that this code need not be rewritten or retested with each new game
application. All code that is used to generate a particular game is contained in a rule
25   EPROM in the rules library 108. Although the step of segregating random number
generator code and transform algorithms has reduced the development time of new
games, further improvements are needed.

4

One significant economic disadvantageous feature with commercial video wagering gaming units that maintains an artificially high price for the systems in the market is the use of unique hardware interfaces in the various manufactured video gaming systems. The different hardware, the different access codes, the different pin couplings,

5   the different harnesses for coupling of pins, the different functions provided from the various pins, and the other various and different configurations within the systems has prevented any standard from developing within the technical field. This is advantageous to the equipment manufacturer, because the games for each system are provided exclusively by a single manufacturer, and the entire systems can be readily obsoleted, so

10   that the market will have to purchase a complete unit rather than merely replacement software, and aftermarket game designers cannot easily provide a single game that can be played on different hardware.

The invention of computerized gaming systems that include a common or "universal" video wagering game controller that can be installed in a broad range of video

15   gaming apparatus without substantial modification to the game controller has made possible the standardization of many components and of corresponding gaming software within gaming systems. Such systems desirably will have functions and features that are specifically tailored to the unique demands of supporting a variety of games and gaming apparatus types, and doing so in a manner that is efficient, secure, and cost-effective to

20   operate.

What is desired is an architecture and method providing a gaming-specific platform that features reduced game development time and efficient game operation, provides security for the electronic gaming system, and does so in a manner that is cost-effective for game software developers, gaming apparatus manufacturers, and gaming

25   apparatus users. An additional advantage is that the use of the platform will speed the review and approval process for games with the various gaming agencies, bringing the games to market sooner.

Summary of the Invention

The present invention in various embodiments provides a computerized wagering game method and apparatus that features an operating system kernel that may include
5   selected device handlers that are disabled or removed. The present invention features a system handler application that is part of the operating system. The system handles loads and executes gaming program objects and features nonvolatile storage that facilitates sharing of information between gaming program objects. The system handler of some embodiments further provides an API library of functions callable from the gaming
10   program shared objects, and facilitates the use of callback functions on change of data stored in nonvolatile storage. A nonvolatile record of the state of the computerized wagering game is stored on the nonvolatile storage, providing protection against loss of the game state due to power loss. The system handler application in various embodiments includes a plurality of handlers, providing an interface to selected hardware
15   and the ability to monitor hardware-related events.

Brief Description of the Figures

Figure 1 shows a computerized wagering game apparatus as may be used to practice an embodiment of the present invention.
20   Figure 2 shows a more detailed structure of program code executed on a computerized wagering game apparatus, consistent with an embodiment of the present invention.

Figure 3 is a diagram illustrating another exemplary embodiment of a universal gaming system according to the present invention having a universal or open operating
25   system.

Figure 4 is a diagram illustrating one exemplary embodiment of a method of converting a gaming system to a gaming system having an open operating system according to the present invention.

Figure 5 is a diagram illustrating one exemplary embodiment of a set of devices used for interfacing with a device driver or handler in an open operating system in a gaming system according to the present invention.

5   Figure 6 is a diagram illustrating one exemplary embodiment of a resource manager used in a gaming system according to the present invention.

Figure 7 is a diagram of a table illustrating one exemplary embodiment of a resource file used in a gaming system according to the present invention.

Figure 8 is a diagram illustrating one exemplary embodiment of converting a cash, coin or token-based gaming system to a cashless gaming system using the universal

10   gaming system according to the present invention.

Figure 9 is a diagram illustrating one exemplary embodiment of configuring a game usable in a gaming system according to the present invention.

Figure 10 is a diagram illustrating another exemplary embodiment of configuring and/or storing a game on a removable media useable in a gaming system according to the

15   present invention.

Figure 11 is a diagram illustrating another exemplary embodiment of a gaming system according to the present invention wherein the game layer is programmable or configurable via a web page at a location remote from the gaming system.

Figure 12 is a diagram illustrating one exemplary embodiment of a web page

20   template used in the gaming system shown in Figure 11.

Figure 13 is a diagram illustrating one exemplary embodiment of nonvolatile memory used in a gaming system according to the present invention, wherein the nonvolatile memory is configured as a RAID system.

25   Detailed Description

In the following detailed description of sample embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific sample embodiments in which the invention may

7

be practiced. These embodiments are described in sufficient detail to enable those skilled
in the art to practice the invention, and it is to be understood that other embodiments may
be utilized and that logical, mechanical, electrical, and other changes may be made
without departing from the spirit or scope of the present invention. The following detailed
5   description is, therefore, not to be taken in a limiting sense, and the scope of the invention
is defined only by the appended claims.

Definitions

For purposes of this disclosure, the following terms have specialized meaning,
10   and are defined below:

"Memory" for purposes of this disclosure is defined as any type of media capable
of read/write capability. Examples of memory are RAM, tape and floppy disc.

"Shared Objects" for purposes of this disclosure are defined as self-contained,
functional units of game code that define a particular feature set or sequence of operation
15   for a game. The personality and behavior of a gaming machine of the present invention
are defined by the particular set of shared objects called and executed by the operating
system. Within a single game, numerous shared objects may be dynamically loaded and
executed. This definition is in contrast with the conventional meaning of a shared object,
which typically provides an API to multiple programs.

20   "Architecture" for purposes of this disclosure is defined as software, hardware or
both.

"Dynamic Linking" for purposes of this disclosure is defined as linking at run
time.

"API" for purposes of this disclosure is an Application Programming Interface.
25   The API includes a library of functions.

"System Handler" for purposes of this disclosure is defined as a collection of code
written to control non-game specific device handlers. Examples of device handlers
include I/O, sound, video, touch screen, nonvolatile RAM and network devices.

8

"Gaming Data Variables" for purposes of this disclosure includes at a minimum any or all data needed to reconstruct the game state in the event of a power loss.

"Game.State File" for purposes of this disclosure is a template for creating a look-up list of information stored in NV RAM.

5   The present invention provides a computerized gaming system method and apparatus that have novel gaming-specific features that improve security, make development of game code more efficient, and do so using an apparatus and software methods that are cost-effective and efficient. The present invention also reduces the amount of effort required to evaluate and review new game designs by gaming regulators,

10   because the amount of code to be reviewed for each game is reduced by as much as 80% over known, machine-specific architecture. The invention provides, in various embodiments, features such as a nonvolatile memory for storing gaming application variables and game state information, provides a shared object architecture that allows individual game objects to be loaded and to call common functions provided by a system

15   handler application, and in one embodiment provides a custom operating system kernel that has selected device handlers disabled.

Figure 1 shows an exemplary gaming system 100, illustrating a variety of components typically found in gaming systems and how they may be used in accordance with the present invention. User interface devices in this gaming system include push

20   buttons 101, joystick 102, and pull arm 103. Credit for wagering may be established via coin or token slot 104, a device 105 such as a bill receiver or card reader, or any other credit input device. A card reader 105 may also provide the ability to record credit information on a user's card when the user has completed gaming, or credit may be returned via a coin tray 106 or other credit return device. Information is provided to the

25   user by devices such as video screen 107, which may be a cathode ray tube (CRT), liquid crystal display (LCD) panel, plasma display, light-emitting diode (LED) display, mechanical reels or wheels or other display device that produces a visual image under control of the computerized game controller. Also, buttons 101 may be lighted to

9

indicate what buttons may be used to provide valid input to the game system at any point
in the game. Still other lights or other visual indicators may be provided to indicate game
information or for other purposes such as to attract the attention of prospective game
users. Sound is provided via speakers 108, and also may be used to indicate game status,
5   to attract prospective game users, to provide player instructions or for other purposes,
under the control of the computerized game controller.

The gaming system 100 further comprises a computerized game controller 111
and I/O interface 112, connected via a wiring harness 113. The universal game controller
111 need not have its software or hardware designed to conform to the interface
10   requirements of various gaming system user interface assemblies, but can be designed
once and can control various gaming systems via the use of machine-specific I/O
interfaces 112 designed to properly interface an input and/or output of the universal
computerized game controller to the harness assemblies found within the various gaming
systems.

15   In some embodiments, the universal game controller 111 is a standard IBM
Personal Computer-compatible (PC compatible) computer. Still other embodiments of a
universal game controller comprise general purpose computer systems such as embedded
controller boards or modular computer systems. Examples of such embodiments include
a PC compatible computer with a PC/104 bus that is an example of a modular computer
20   system that features a compact size and low power consumption while retaining PC
software and hardware compatibility. The universal game controller 111 provides all
functions necessary to implement a wide variety of games by loading various program
code on the universal controller, thereby providing a common platform for game
development and delivery to customers for use in a variety of gaming systems. Other
25   universal computerized game controllers consistent with the present invention may
include any general-purpose computers that are capable of supporting a variety of gaming
system software, such as universal controllers optimized for cost effectiveness in gaming
applications or that contain other special-purpose elements yet retain the ability to load

and execute a variety of gaming software. Examples of special purpose elements include elements that are heat resistant and are designed to operate under less than optimal environments that might contain substances such as dust, smoke, heat and moisture. Special purpose elements are also more reliable when used 24 hours per day, as is the

5   case with most gaming applications.

The computerized game controller of some embodiments is a computer running an operating system with a gaming application-specific kernel. In further embodiments, a game engine layer of code executes within a non-application specific kernel, providing common game functionality. The gaming program shared object in such embodiments is

10   therefore only a fraction of the total code, and relies on the game engine layer and operating system kernel to provide a library of gaming functions. A preferred operating system kernel is the public domain Linux 2.2 kernel available on the Internet. Still other embodiments will have various levels of application code, ranging from embodiments containing several layers of game-specific code to a single-layer of game software

15   running without an operating system or kernel but providing its own computer system management capability.

Figure 2 illustrates the structure of one exemplary embodiment of the invention, as may be practiced on a computerized gaming system such as that of Figure 1. The invention includes an operating system 300, including an operating system kernel 201 and

20   a system handler application 202. An operating system kernel 201 is first executed, after which a system handler application 202 is loaded and executed. The system handler application in some embodiments may load a gaming program shared object 203, and may initialize the game based on gaming data variables stored in nonvolatile storage 204. In some embodiments, the gaming data variables are further loaded into a Game.State

25   data file or other data storage device 205, which reflects the data stored in nonvolatile storage 204. The nonvolatile RAM (NV-RAM) according to the invention has read/write capability. The gaming program object in some embodiments calls separate API

11

functions 206, such as sound functions that enable the gaming apparatus to produce sound effects and music.

The OS kernel 201 in some embodiments may be a Linux kernel, but in alternate embodiments may be any other operating system providing a similar function. The Linux 2.2 operating system kernel in some further embodiments may be modified for adaptation to gaming architecture. Modifications may comprise erasing or removing selected code from the kernel, modifying code within the kernel, adding code to the kernel or performing any other action that renders the device handler code inoperable in normal kernel operation. By modifying the kernel in some embodiments of the invention, the function of the computerized gaming apparatus can be enhanced by incorporating security features, for example. In an embodiment, all modifications to the kernel are modular.

For example, as described in my co-pending application Serial No. _____, entitled "Encryption in a Secure Computerized Gaming System" filed on the same date as the present application, several functions are incorporated into the kernel to verify that the operating system and shared object code have not changed, and that no new code has been incorporated into the operating system code or shared object code.

In one embodiment, the kernel is modified so that it executes user level code out of ROM. The use of the Linux operating system lends itself to this application because the source code is readily available. Other operating systems such as Windows and DOS are other suitable operating systems.

Embodiments of the invention include hard real time code 310 beneath the kernel providing real time response such as fast response time to interrupts. The hard real time code 310 is part of the operating system in one embodiment.

In an embodiment of the invention, all user interface peripherals such as keyboards, joysticks and the like are not connected to the architecture so that the operating system and shared objects retain exclusive control over the gaming machine. In another embodiment, selected device handlers are disabled so that the use of a keyboard, for example, is not possible. It is more desirable to retain this functionality so that user

peripherals can be attached to service the machine. It might also be desirable to attach
additional user peripherals such as tracking balls, light guns, light pens and the like.

In another embodiment, the kernel is modified to zero out all unused RAM. This
function eliminates code that has been inserted unintentionally, such as through a Trojan
5   horse, for example.

In one embodiment, the kernel and operating system are modified to hash the
system handler and shared object or gaming program object code, and to hash the kernel
code itself. These functions in different embodiments are performed continuously, or at a
predetermined frequency.

10 The system handler application is loaded and executed after loading the operating
system, and manages the various gaming program shared objects. In further
embodiments, the system handler application provides a user Application Program
Interface (API) 206, that includes a library of gaming functions used by one or more of
the shared objects 210. For example, the API in one embodiment includes functions that
15 control graphics, such as color, screen commands, font settings, character strings, 3-D
effects, etc. The device handlers 210 are preferably handled by an event queue 320. The
event queue schedules the event handlers in sequence. The shared object 203 calls the
APIs 206 in one embodiment. The system handler application 202 in various
embodiments also manages writing of data variables to the "game.state" file 205 stored in
20 the nonvolatile storage 204, and further manages calling any callback functions associated
with each data variable changed.

The system handler 202 application of some embodiments may manage the
gaming program shared objects by loading a single object at a time and executing the
object. When another object needs to be loaded and executed, the current object may
25 remain loaded or can be unloaded and the new object loaded in its place before the new
object is executed. The various shared objects can pass data between objects by storing
the data in nonvolatile storage 204, utilizing a game.state data storage device 205. For
example, a "game.so" file may be a gaming program object file that is loaded and

executed to provide operation of a feature set of a computerized wagering game, as a
"bonus.so" gaming program object file is loaded and executed to provide a feature set of
the bonus segment of play. Upon changing from normal game operation to bonus, the
bonus.so is loaded and executed upon loading. Because the relevant data used by each
5   gaming program object file in this example is stored in nonvolatile storage 204, the data
may be accessed as needed by whatever gaming program object is currently loaded and
executing.

     The system handler application in some embodiments provides an API that
comprises a library of gaming functions, enabling both easy and controlled access to
10   various commonly used functions of the gaming system. Providing a payout in the event
of a winning round of game play, for example, may be accomplished via a payout
function that provides the application designer's only access to the hardware that pays out
credit or money. Restrictions on the payout function, such as automatically reducing
credits stored in nonvolatile storage each time a payout is made, may be employed in
15   some embodiments of the invention to ensure proper and secure management of credits
by the computerized gaming system. The functions of the API may be provided by the
developer as part of the system handler application, and may be a part of the software
provided in the system handler application package. The API functions may be updated
as needed by the provider of the system handler application to provide new gaming
20   functions as desired. In some embodiments, the API may simply provide functions that
are commonly needed in gaming, such as computation of odds or random numbers, an
interface to peripheral devices, or management of cards, reels, video output or other
similar functions.

     The system handler application 202 in various embodiments also comprises a
25   plurality of device handlers 210, that monitor for various events and provide a software
interface to various hardware devices. For example, some embodiments of the invention
have handlers for nonvolatile memory 212, one or more I/O devices 214, a graphics
engine 216, a sound device 218, or a touch screen 220. Also, gaming-specific devices

such as a pull arm, credit receiving device or credit payout device may be handled via a
device handler 222. Other peripheral devices may be handled with similar device
handlers, and are to be considered within the scope of the invention. In one embodiment,
the device handlers are separated into two types. The two types are: soft real time 210A
5   and regular device handlers 210B. The two types of device handlers operate differently.
The soft real time handler 210A constantly runs and the other handler 210B runs in
response to events.

The nonvolatile storage 204 used to store data variables may be a file on a hard
disc, may be nonvolatile memory, or may be any other storage device that does not lose
10 the data stored thereon upon loss of power. In one embodiment the nonvolatile storage in
battery-backed RAM. The nonvolatile storage in some embodiments may be encrypted to
ensure that the data variables stored therein cannot be corrupted. Some embodiments
may further include a game.state file 205, which provides a look-up table for the game
data stored in nonvolatile storage 204. The game.state file is typically parsed prior to
15 execution of the shared object file. The operating system creates a map of NVRAM by
parsing the game.state file. The look-up table is stored in RAM. This look-up table is
used to access and modify game data that resides in NVRAM 204. This game data can
also be stored on other types of memory.

In some embodiments, a duplicate copy of the game data stored in NVRAM 204
20 resides at another location in the NVRAM memory. In another embodiment, a duplicate
copy of the game data is copied to another storage device. In yet another embodiment,
two copies of the game data reside on the NVRAM and a third copy resides on a separate
storage device. In yet another embodiment, three copies of the game data reside in
memory. Extra copies of the game data are required by gaming regulations in some
25 jurisdictions.

Data written to the game state device must also be written to the nonvolatile
storage device, unless the game state data device is also nonvolatile, to ensure that the
data stored is not lost in the event of a power loss. For example, a hard disc in one

embodiment stores a game.state file that contains an unencrypted and nonvolatile record
of the encrypted data variables in nonvolatile storage flash programmable memory (not
shown). Data variables written in the course of game operation are written to the
game.state file, which may be encrypted and stored in the nonvolatile storage 204, upon
5   normal shutdown. Loss of power leaves a valid copy of the most recent data variables in
the game.state file, which may be used in place of the data in NVRAM in the event of an
abnormal shutdown.

In an alternate embodiment, a game state device 205 such as a game.state file
stored on a hard disc drive provides variable names or tags and corresponding locations in
10   nonvolatile storage 204, in effect, providing a variable map of the nonvolatile storage. In
one such embodiment, the nonvolatile storage may then be parsed using the data in the
game state file 205, which permits access to the variable name associated with a
particular nonvolatile storage location. Such a method permits access to and handling of
data stored in nonvolatile storage using variable names stored in the game state file 205,
15   allowing use of a generic nonvolatile storage driver where the contents of the nonvolatile
storage are game-specific. Other configurations of nonvolatile storage such as a single
nonvolatile storage are also contemplated, and are to be considered within the scope of
the invention.

Callback functions that are managed in some embodiments by the system handler
20   application 202 are triggered by changing variables stored in NVRAM 204. For each
variable, a corresponding function may be called that performs an action in response to
the changed variable. For example, every change to a "credits" variable in some
embodiments calls a "display_credits" function that updates the credits as displayed to the
user on a video screen. The callback function may be a function provided by the current
25   gaming program shared object or can call a different gaming program object.

The gaming program's shared objects in some embodiments of the invention
define the personality and function of the game. Program objects provide different game
functions, such as bookkeeping, game operation, game setup and configuration functions,

16

bonus displays and other functions as necessary. The gaming program objects in some embodiments of the invention are loaded and executed one at a time, and share data only through NVRAM 204 or another game data storage device. The previous example of unloading a game.so gaming program object and replacing it with a bonus.so file to

5   perform bonus functions is an example of such use of multiple gaming program shared objects.

Each gaming program object may require certain game data to be present in NVRAM 204, and to be usable from within the executing gaming program shared object 203. The game data may include meter information for bookkeeping, data to recreate

10 game on power loss, game history, currency history, credit information, and ticket printing history, for example. These files do not include operable code or functions.

The operating system of the present application is not limited to use in gaming machines. It is the shared object library rather than the operating system itself that defines the personality and character of the game. The operating system of the present

15 invention can be used with other types of shared object libraries for other purposes.

For example, the operating system of the present invention can be used to control networked on-line systems such as progressive controllers and player tracking systems. The operating system could also be used for kiosk displays or for creating "picture in picture" features in gaming machines. A gaming machine could be configured so that a

20 video slot player could place a bet in the sports book, then watch the sporting event in the "picture in picture" feature while playing his favorite slot game.

The present invention provides a computerized gaming apparatus and method that provides a gaming-specific platform that features reduced game development time and efficient game operation via the use of a system handler application that can manage

25 independent gaming program objects and gaming-specific API, provides game functionality to the operating system kernel, provides security for the electronic gaming system via the nonvolatile storage and other security features of the system, and does so in an efficient manner that makes development of new software games relatively easy.

Production and management of a gaming apparatus is also simplified, due to the system handler application API library of gaming functions and common development platform provided by the invention.

       Figure 3 is a diagram illustrating one exemplary embodiment of a gaming system

5   400 according to the present invention including universal operating system 300. As previously described herein, game layer 402 include gaming program shared objects 203 which are specific to the type of game being played on gaming system 400. Exemplary game objects or modules include paytable.so 406, help.so 408 and game.so 410. Game layer 402 also includes other game specific independent modules 412. Game engine 404

10   provides an interface between game layer 402 and universal operating system 300. The game engine 404 provides an additional application programming interface to the game layer application. The game engine automates core event handling for communicating with the game operating system 300, and which are not configurable via the specific game layer game code. The game engine 404 also provides housekeeping and game state

15   machine functions. The game layer objects 203 and/or modules 406 may also directly call user API 206.

       As previously described herein universal operating system 300 is an open operating system which allows for conversion of the gaming system 400 into different types of games, and also allows for future expandability and upgrading of associated

20   hardware in the gaming system 400 due to its open architecture operating system.

       In operating system 300, device handlers 210 provide the interface between the operating system 300 and external gaming system input and output devices, such as a button panel, bill acceptor, coin acceptor, mechanical arm, reels, speaker, tower lights, etc. Each device handler 210 is autonomous to the other. The device handlers or drivers

25   210 operate as protocol managers which receive information from a gaming system device (typically in the gaming system device protocol) and converts the information to a common open operating system protocol usable by operating system 300. Similarly, the device drivers or handlers 210 receive information from the open operating system and

convert the information to a gaming device specific protocol. The specific device handlers or drivers used are dependent upon what game you are using, and may be loadable or unloadable as independent, separate objects or modules. The exemplary embodiment shown includes total I/O device handler 414, sound device handler 416,

5   serial device handler 418, graphics device handler 420, memory manager device handler 422, NVRAM device handler 424, protocols device handler 426, resource manager device handler 428 and network device handler 430. Other suitable device handlers for adapting the operating system 300 to other gaming systems will become apparent to one skilled in the art after reading the present application.

10   Figure 4 is a diagram illustrating one exemplary embodiment of a method of converting an existing gaming operating system to a gaming system 400 having an open operating system 300 according to the present invention. The gaming system 400 according to the present invention. The gaming system 400 according to the present invention is suitable for converting both video based gaming systems and also

15   electrical/mechanical based operating system, such as a mechanical reel based slot machine. Once the existing game operating system has been changed over to a universal gaming system 400 having a universal operating system 300 according to the present invention, the type of game itself may be changed via changing out the game specific code in the game layer 402.

20   At 450, the existing game operating system is removed from the game. The existing game operating system is typically a proprietary operating system consisting of computer hardware and software which is specific to the game being changed out. At 452, a universal gaming system 402 including an open operating system 300 is installed in the game. At 454, functional interfaces are provided between the open operating

25   system and the existing gaming system devices. At 456, a game specific program (i.e., game layer 402) is installed in the universal gaming system. The game specific program is configured to communicate with the open operating system 300.

In one exemplary embodiment, the gaming system according to the present
invention is used in a mechanical reel-based slot machine, either in a new slot machine or
in converting an existing slot machine to an open operating system according to the
present invention. Exemplary conventional reel-based slot machines include an IGT S-
5   plus slot machine or a Bally slot machine.

Figure 5 is a diagram illustrating one exemplary embodiment of I/O devices which
must be functionally interfaced within adopting gaming system 402 to a reel-based game.
The exemplary embodiment shown includes devices which interface with a digit I/O
device driver. In one embodiment, input devices 462 includes a button panel device 466,
10   a mechanical arm device 468, a bill acceptor device 470, and a coin acceptor device 472.
Each of the input devices 462 receives information from the specific game devices and
provides the information to the gaming system 400 via the I/O device driver.

Output devices 464 receive information from operating system 300 which
provides an output via the I/O device driver to gaming devices 464. In the example
15   shown, output devices 464 include reels device 474 which receives an output to the
stepper motors controlling the reels. Credit displays device 466 which receives an output
to the LED driven credit displays. Speaker device 478 which receives a sound output to
the game speakers. On-line system protocol devices 480 which is a communication
interface between the open operating system 300 and the game on-line system. Tower
20   lights devices 42 which receives an interface between the open operating system 300 and
the game tower lights.

Figure 6 is a diagram illustrating one exemplary embodiment of a resource
manager used in a gaming system according to the present invention. The resource
manager 500 is a software module which receive a resource configuration file 502 and
25   stores it in memory 504. In one aspect, memory 504 is stored in nonvolatile memory,
which in one embodiment is flash memory. The resource manager parses the resource
configuration file and stores individual resources in memory for fast recall.

In one embodiment, the resource manager 500 stores the file 506 in the form of a lookup table. In one preferred embodiment, the resource manager reads the configuration files at startup, parses the configuration files and stores them in memory 504. The resource manager file 506 may then be accessed by the rest of the operating system 300

5   software applications. The resource manager operates to map digital I/O lines, com ports, game specific resources, kernal modules to load, etc.

Figure 7 is a diagram of a table illustrating one exemplary embodiment of a portion of a resource file 506 according to the present invention. The resource manager 500 operates to map the input/output (I/O) line to the operating system resources. Instead

10 of changing pin locations for different games, the resource manger provides for mapping of I/O lines via software. In one aspect, 64, I/O (X8) lines are mapped to the various operating system resources. In one aspect, the I/O line at PIN1 510 is mapped to resource R20 512; and PIN2 514 is mapped to resource R3 516; PIN3 518 is mapped to resource R38 520; PIN4 522 is mapped to resource R10 524; PIN5 526 is mapped to resource R11

15 528; PIN6 530 is mapped to resource R12 532; PIN7 534 is mapped to resource R13 536; and PINN 538 is mapped to resource R51 540, etc.

The gaming system 400 according to the present invention is adaptable for use as a cashless gaming system. As such, it is useable for converting existing coin-based or token-based gaming systems into a cashless gaming system.

20 Figure 8 is a diagram illustrating one exemplary embodiment of converting cash, coin, or token-based gaming system to a cashless gaming system using the universal gaming system 400 according to the present invention. References also made to Figures 1-7 previously described herein. A card reader or coupon acceptor 550 and ticket printer 552 are added to the game. The card reader 550 and ticket printer 552 are easily

25 adaptable to interface with the gaming system 400 according to the present invention. In particular, card reader device driver 554 is added to open operating system 300 to enable card reader 550 to communicate with the operating system. Similarly, a ticket printer device driver 556 is added to the operating system 300 in order to allow ticket printer 552

21

to communicate with the operating system. For example, an existing cash-based reel slot machine can be converted according to the present invention to a cashless gaming system. The card reader 550 can operate to read credit cards, magnetic strip based cards, or accept coupons which includes credits such as promotional gaming credits received from
5 a casino. The card or coupons may be obtainable from a central or kiosk location. Once play is complete on the gaming system 400, the ticket printer 556 is operable to print a ticket representative of the amount of credits or money won on the gaming system. The ticket 560 may then be used as a card or coupon in another gaming system, or alternatively, may be turned in at a kiosk or central location for money.

10   Figure 9 is a diagram illustrating another exemplary embodiment of the gaming system 400 according to the present invention. Due to the open operating system 300, game layer 402 may be configurable remote from the gaming system 400, such as on a remote computer or laptop computer 580. Game layer 402 is constructed into game objects or modules 302. As such, templates for specific types of games are configured to
15 allow a game programmer to specify game specific configurable options from a remote computer 580. In another aspect, game specific modules are created on the remote computer 580. The game layer is then assembled and transferred into memory 582. In one aspect, memory 582 is nonvolatile memory located in the gaming system 400. In one aspect, the nonvolatile memory is flash memory. In one exemplary embodiment, the
20 flash memory is a "Disk on a Chip", wherein the game layer 402 is downloaded from the remote computer 580 onto the disk on a chip 582.

            Figure 10 is a diagram illustrating another exemplary embodiment of programming and/or configuring a game layer at a location remote from the gaming system 400. In this embodiment, game layer 402 is programmed or configured on remote
25 computer 580. After completion of configuring and/or programming game layer 402, the game layer 402 is transferred via remote computer 580 to a removable media 584. In one preferred embodiment, the removable media is a flash memory card, and more preferably is a CompactFlash card. In one aspect, the flash memory card plugs into remote

computer 580 via a PCMCIA slot. Suitable flash memory cards (i.e., a CompactFlash card) are commercially available from memory manufacturers, including SanDisk and Kingston.

The removable media 584 is removed from remote computer 580 and inserted in
5   gaming system 400. In one aspect, removable media 584 can be "hot-inserted" directly into the controller board of gaming system 400. The removable media 584 contains game layer 402 including the game specific code and program files. As such, removable media 584 remains inserted into gaming system 400 during operation of the gaming system. In an alternative embodiment, the game layer 402 can be transferred (e.g., via a memory
10 download) from removable media 584 to memory inside of gaming system 400.

In one embodiment, the removable media 584 is maintained in gaming system 400 during operation of the gaming system. As such, the gaming system program files may be verified for authenticity by gaming officials by simply removing the removable media 584 and inserting it in a computer or controller used for verifying/authenticating game
15 code, indicated at 586.

Figure 11 is another exemplary embodiment of a gaming system according to the present invention wherein the game layer is programmable or configurable at a location remote from the gaming system 400. In this embodiment, game layer 402 is configurable over a network based communication system. In one embodiment, network based system
20 600 includes a user interface 602, network or network communication link 604, and controller 606. Controller 606 is configured to communicate with user 610 via network 604. In particular, centralized controller 606 includes web server 612. User 610 accesses web server 612 via user interface 602, and downloads a web page suitable for configuring a game layer. In one aspect, the web page includes game specific game templates 608,
25 which are utilized for inputting specific user configurations for individual games. Once a game template 608 has been configured, the game template is transferred to controller 606 via network 604. Controller 606 receives the configuration information associated with game template 608 and assembles game layer or program 402 using the

configuration information. Game layer or program 402 can now be downloaded into memory in gaming system 400 for use by gaming system 400 including the game specific configurable options determined by user 610.

The system 600 also allow other user interfaces 614 for configuring games which
5   may be assembled by controller 606 for use in other gaming systems. Alternatively, other user interface 614 may be representative of a gaming official checking the game 402 and authorizing use of the game 402 and gaming system 400. As such, the game 402 may be transferred to the gaming system 400 via controller 606, or via a communication link with user interface 64, which may be a direct connection or may be a network. Alternatively,
10 game layer 402 may be transferred from controller 606 or user interface 614 by putting it on a flash memory device (e.g., Disk on a Chip or CompactFlash card) and physically inserted into gaming system 400.

Network 604, as used herein, is defined to include an internet network (e.g., the Internet), intranet network, or other high-speed communication system. In one preferred
15 embodiment, network 44 is the Internet. While the exemplary embodiment and this detailed description refers to the use of web pages on the Internet network, it is understood that the use of other communication networks or next generation communication networks or a combination of communication networks (e.g., and intranet and the Internet) are within the scope of the present invention. The assembly of
20 configuration information received from user interface 602 can be assembled into game layer 402 using hardware via a microprocessor, programmable logic, or state machine, in firmware, and in software within a given device. In one aspect, at least a portion of the software programming is web-based and written in HTML and JAVA programming languages, including links to the web pages for data collection, and each of the main
25 components communicate via network 604 using a communication bus protocol. For example, the present invention may or may not use a TC/IP protocol suite for data transport. Other programming languages and communication bus protocols suitable for

use with the system 600 according to the present invention will become apparent to those skilled in the art after reading the present application.

Figure 12 is a diagram illustrating one exemplary embodiment of web page game templates used in the system shown in Figure 11. Template 1 is shown at 622 and
5 Template 2 is shown at 624. In one embodiment, upon accessing controller 606 via user interface 602, user 610 selects a game type that the user 610 would like to either program or configure. Based on the game type 626, a template appears at user interface 602 for that game type which allows the user to specify game configurable options, indicated at 628. The controller then operates to assemble the game layer or game programs 402
10 based on the information received via the game template. The configurable options may include any type of game specific configurable options, such as game colors, game sound, percentage payouts, game options, etc.

Figure 13 is a diagram illustrating one exemplary embodiment of nonvolatile RAM used in a gaming system 400 according to the present invention, wherein the
15 nonvolatile RAM is configured as a redundant memory system. In one exemplary embodiment shown, the nonvolatile RAM is configured as a RAID system. In the hard disk drive industry, RAID (short for redundant array of independent disks) systems employ two or more disk drives in combination for improved disk drive fault tolerance and disk drive performance. RAID systems stripe a user's data across multiple hard
20 disks. When accessing data, the RAID system allows all of the hard disks to work at the same time, providing increase in speed and reliability.

A RAID system configuration as defined by different RAID levels. The different RAID levels range from LEVEL 0 which provides data striping (spreading out of data blocks of each file across multiple hard disks) resulting in improved disk drive speed and
25 performance but no redundancy. RAID LEVEL 1 provides disk mirroring, resulting in 100 percent redundancy of data through mirrored pairs of hard disks (i.e., identical blocks of data written to two hard disks). Other drive RAID levels provide variations of data

striping and disk mirroring, and also provide improved error correction for increased performance and fault tolerance.

In Figure 13, one exemplary embodiment of RAID data storage system used in a gaming system 400 according to the present invention is generally shown at 630. The

5   RAID storage system 630 includes a controller or control system 632 and multiple nonvolatile RAM data storage units, indicated as RAMA 634 and RAMB 636. In one aspect, RAMA 634 and RAMB 636 each include a backup power system PWR 638 and PWR 640. In one aspect, backup power systems PWR 638 and PWR 640 are battery backup systems. RAMA 634 and RAMB 636 are configured to communicate with

10 control system 632 as a redundant array of storage devices. Preferably, nonvolatile memory RAMA 634 and nonvolatile memory RAMB 636 are configured similar to a RAID level configuration used in the disk drive industry (i.e., as a "mirrored pair"). Nonvolatile memory RAMA 634 and nonvolatile memory RAMB 636 communicate with control system 632 via communication bus 638, using a communication bus protocol.

15 One exemplary embodiment of a communication bus suitable for use as communication bus 638 is an industry standard ATA or uniform serial bus (USB) communication bus. Control system 632 includes a microprocessor based data processing system or other system capable of performing a sequence of logical operations. In one aspect, control system 632 is configured to operate the RAID system 630 nonvolatile memories RAMA

20 634 and RAMB 636 as a mirrored pair. As such, read/write to nonvolatile memory RAMA 634 are mirrored to nonvolatile RAMB 636, providing redundancy of crucial gaming specific data stored in nonvolatile memory RAMA 634 and RAMB 636. Alternatively, the nonvolatile memory RAMA 634 and nonvolatile memory RAMB 636 may be configured to communicate with control system 632 similar with other RAID

25 storage system levels, such as RAID LEVEL 0, RAID LEVEL 2, RAID LEVEL 3, RAID LEVEL 4, RAID LEVEL 5, RAID LEVEL 6, etc. Further, the RAID system 630 may include more than the two nonvolatile memories RAMA 634 and RAMB 636 shown.

Although specific embodiments have been illustrated and described herein, it will
be appreciated by those of ordinary skill in the art that any arrangement which is
calculated to achieve the same purpose may be substituted for the specific embodiments
shown. This application is intended to cover any adaptations or variations of the
5   invention. It is intended that this invention be limited only by the claims, and the full
scope of equivalents thereof.

WHAT IS CLAIMED IS:

1.   A computerized wagering game apparatus, comprising:
        a computerized game controller having a processor, memory, and
nonvolatile storage and is operable to control the computerized wagering game; and
                an operating system comprising: a system handler application operable to
dynamically link with at least one gaming program object; and
                an operating system kernel that executes the system handler application.

2.   The computerized wagering game apparatus of claim 1, wherein the
system handler application comprises a plurality of device handlers.

3.   The computerized wagering game apparatus of claim 1, wherein the
system handler application comprises software having the ability when executed to:
                load a gaming program shared object; and
                execute the new gaming program shared object.

4.   The computerized wagering game apparatus of claim 1, wherein game data
modified by the gaming program objects is stored in nonvolatile storage.

5.   The computerized wagering game apparatus of claim 4, wherein changing
game data in nonvolatile storage causes execution of a corresponding callback function in
the system handler application.

6.   The computerized wagering game apparatus of claim 1, wherein the
computerized game controller comprises an IBM PC-compatible controller.

7.   The computerized wagering game apparatus of claim 1, wherein the operating system kernel is a Linux operating system kernel.

8.   The computerized wagering game apparatus of claim 7, wherein the Linux operating system kernel is modified.

9.   The computerized wagering game apparatus of claim 8, wherein the kernel has at least one modification wherein each modification is selected from the group consisting of: 1) accessing user level code from ROM, 2) executing from ROM, 3) zero out unused RAM, 4) test and/or hash the kernel, and 5) disabling selected device handlers.

10.   The computerized wagering game apparatus of claim 9, wherein the modifications are modular.

11.   The computerized wagering game apparatus of claim 1, wherein the system handler application comprises an API with functions callable from the gaming program objects.

12.   The computerized wagering game apparatus of claim 1, wherein the system handler further comprises an event queue.

13.   A method of managing data in a computerized wagering game apparatus via a system handler application, comprising:

            loading a shared object,
            executing the shared object, and
            accessing and storing game data in nonvolatile storage.

14.   The method of claim 13, and further comprising the step of unloading the first program object, and further comprising loading a second program object.

15.   The method of claim 13, further comprising executing a corresponding callback function upon alteration of game data in nonvolatile storage.

16.   A computerized wagering game system controlled by a general-purpose computer, comprising an operating system kernel that is customized for gaming use.

17.   The computerized wagering game system of claim 16, wherein the kernel is customized in at least one way; selected from the group: 1) accessing user level code from ROM, 2) executing from ROM, 3) zero out unused RAM, 4) test and/or hash the kernel, and 5) disabling selected device handlers.

18.   A computerized wagering game system controlled by a general-purpose computer comprising nonvolatile storage that stores game data, such that loss of power does not result in loss of the state of the computerized wagering game system.

19.   A gaming machine operating system, comprising a processor and memory and is operable to control the computerized wagering game, wherein the memory contains a plurality of shared objects and a system handler, and the system handler is adapted to execute at least one shared object called from memory.

20.   The operating system of claim 19, further comprising nonvolatile storage, wherein game data stored in nonvolatile storage is retained during a gaming machine power down.

21.   A machine-readable medium with instructions thereon, the instructions
when executed operable to cause a computer to:

      load a first program shared object,

      execute a first program shared object,

      store game data in nonvolatile storage, such that a second program object
later loaded can access the data variables in nonvolatile storage,

      unload the first program shared object, and

      load the second program shared object.

22.   The machine-readable medium of claim 21, with further instructions
operable when executed to cause a computer to execute a corresponding callback function
upon alteration of game data in the nonvolatile storage.

23.   The machine-readable medium of claim 22, with further instructions
operable when executed to cause a computer to manage events via the system handler
application.

24.   A machine-readable medium with instructions thereon, the instructions
when executed operable to cause a computer to manage at least one gaming program
object via a system handler application, such that a single gaming program object is
executed at any one time, wherein gaming program objects are operable to share game
data in nonvolatile storage.

25.   The machine-readable medium of claim 21, wherein only one gaming
program object executes at any one time.

26.   The machine-readable medium of claim 21, with further instructions operable when executed to cause a computer to provide functions through an API that comprises a part of the system handler application.

27.   A machine-readable medium with instructions thereon, the instructions when executed are operable to store game data in nonvolatile storage, such that the state of the computerized wagering game system is maintained when the machine loses power.

28.   A gaming machine architecture, comprising an operating system, and a plurality of shared objects; wherein each shared object describes game personality in a selected mode.

29.   The gaming machine architecture of claim 28, wherein the operating system comprises an IBM PC-based operating system.

30.   The gaming machine architecture of claim 28, wherein the operating system comprises a system handler.

31.   The gaming machine architecture of claim 30, wherein the system handler comprises a plurality of device handlers.

32.   The gaming machine architecture of claim 30, wherein the system handler comprises an event queue.

33.   The gaming machine architecture of claim 30, wherein the system handler comprises a plurality of API callable functions.

34.   The apparatus of claim 1, wherein the system handler comprises an event queue that determines the order of execution of each specified device handler.

35.   The apparatus of claim 1, wherein the system handler comprises an API having a library of functions.

36.   The apparatus of claim 34, wherein the event queue is capable of queuing on a first come, first serve basis.

37.   The apparatus of claim 34, wherein the event queue is capable of queuing using more than one criteria.

38.   The apparatus of claim 1, wherein the system handler and kernel work in communication to hash the system handler code and operating system kernel code.

39.   A universal operating system, comprising a system handler; and an operating system kernel.

40.   The operating system of claim 39, and further comprising a plurality of APIs.

41.   The operating system of claim 39, and further comprising an event queue.

42.   The operating system of claim 39, wherein the system handler comprises a plurality of device handlers.

43.   The operating system of claim 39, wherein the operating system kernel is customized for gaming purposes.

44.   The operating system of claim 43, wherein the kernel is customized utilizing a method selected from the group consisting of: 1) Accessing user level code from ROM, 2) executing from ROM, 3) zero out unused RAM, 4) test and/or hash the kernel, and 5) disabling selected device handlers.

45.   The operating system of claim 39, wherein the system is used to control a networked on-line system.

46.   The operating system of claim 39, wherein the system is used to control a progressive meter.

47.   A method of modifying an operating system kernel, comprising at least one modification to obtain functionality selected from the group consisting of:

1)   accessing user level code from ROM;

2)   executing user level code from ROM;

3)   zeroing out unused RAM;

4)   testing and/or hashing the kernel; and

5)   disabling selected device handlers.

48.   A method of converting a game to operate on a universal gaming system comprising:

        removing a game operating system from the game, the game operating system including hardware and software;

        installing a universal gaming system in place of the game operating system, the universal gaming system including a game program layer, an open operating system, and a game controller for running the game program layer on the open operating system;

34

providing functional interfaces between the universal gaming system and game devices; and

installing a game specific program in the game program layer configured to operate with the open operating system.

49.   The method of claim 48, comprising:

defining the open operating system to include a system application handler, wherein the functional interface between the gaming system and the game devices is accomplished via the system application handler.

50.   The method of claim 49, comprising:

configuring the system handler application to include one or more device handlers for interfacing with the game devices, wherein at least one of the device handlers operates as a protocol manager between the games device and the open operating system.

51.   The method of claim 48, comprising:

defining the open operating system to include an operating system kernal the executes the system handler application.

52.   The method of claim 48, comprising:

defining the game program layer to include one or more gaming program shared objects.

53.   The method of claim 52, wherein the gaming program shared objects are specific to the type of game played on the universal gaming system.

54.   The method of claim 53, comprising:

changing the type of game played on the universal gaming system by changing the shared objects.

55.   The method of claim 48, comprising:

configuring the game program layer to operate the game as a slot machine.

56.   The method of claim 55, comprising:

defining the slot machine to be a mechanical reel-based slot machine.

57.   The method of claim 48, comprising:

configuring the open operating system to include a resource manager for mapping game specific resources.

58.   The method of claim 57, wherein mapping game specific resources includes parsing a configuration file, mapping operating system resources based on the configuration file, and storing the resource map in memory.

59.   The method of claim 58, wherein mapping operating system resources based on the configuration file includes mapping input/output lines to system resources.

60.   The method of claim 48, comprising converting the game to a cashless gaming system including defining the open operating system to include a system application handler, wherein the functional interface between the gaming system and the game devices is accomplished via the system application handler, and configuring the system handler application to include one or more device handlers for interfacing with the game devices, including installing a card reader device handler; and installing a card reader in communication with the card reader device handler.

61.   The method of claim 60, including configuring the system handler application to include a ticket printer device handler; and installing a ticket printer in communication with the ticket printer device handler.

62.   The method of claim 60, comprising configuring the game program layer to include a cashless gaming feature.

63.   A method of converting a mechanical reel slot machine game to operate on a universal gaming system comprising:

removing a slot machine game operating system from the game, the game operating system including hardware and software;

installing a universal gaming system in place of the slot machine game operating system, the universal gaming system including a game program layer, an open operating system, and a game controller for running the game program layer on the open operating system;

providing functional interfaces between the universal gaming system and slot machine game devices; and

installing a game specific program in the game program layer configured to operate with the open operating system.

64.   The method of claim 63, comprising:

defining the open operating system to include a system application handler, wherein the functional interface between the gaming system and the slot machine game devices is accomplished via the system application handler.

65.   The method of claim 64, comprising:

configuring the system handler application to include one or more device handlers for interfacing with the slot machine game devices, wherein at least one of the device handlers operates as a protocol manager between the slot machine games device and the open operating system.

66.   The method of claim 65, comprising:

configuring an I/O device handler to interface with slot machine input devices and slot machine output devices.

67.   The method of claim 66, comprising:

defining the slot machine input devices to include a mechanical arm, button acceptor and coin acceptor.

68.   The method of claim 66, comprising:

defining the slot machine output devices to include slot machine reels, credit displays, and speakers.

69..   The method of claim 65, comprising:

converting the mechanical reel slot machine game to a cashless gaming system via the system handler application, including providing a card reader device handler, and installing a card reader in communication with the card reader device handler.

70.   The method of claim 69, comprising:

providing a ticket printer device handler, and installing a ticket printer in communication with the ticket printer device handler.

71.   A method of configuring a game program layer for a universal gaming system configured for a game program layer and an open operating system, the method comprising:

configuring the game program layer on a computer remote from the gaming system; and

downloading the game program layer into the universal gaming system.

72.   The method of claim 71, comprising:

defining a game template; and

configuring the game program layer using the game template.

73.   The method of claim 71, comprising:

storing the game program on a removable media card.

74.   The method of claim 73, comprising defining the removable media to be flash memory.

75.   The method of claim 73, comprising defining the flash memory to be a CompactFlash card.

76.   The method of claim 73, comprising plugging the removable media card into the gaming system, and running the game program layer via the open operating system from the removable media card.

78.   The method of claim 76, comprising authenticating the game program layer by plugging the removable media card into an authenticating system.

79.   A network based method of configuring a game program layer for a
universal gaming system configured for a game program layer and an open operating
system, the method comprising:

       defining a user interface;

       configuring the game program layer via the user interface remote from the
gaming system;

       defining a controller having a web server;

      downloading the game program layer into the controller via a network;

       transferring the game program layer to the universal gaming system.

80.   The method of claim 79, comprising configuring the game program layer
via a web page template at the user interface.

81.   The method of claim 79, comprising configuring the controller to be part
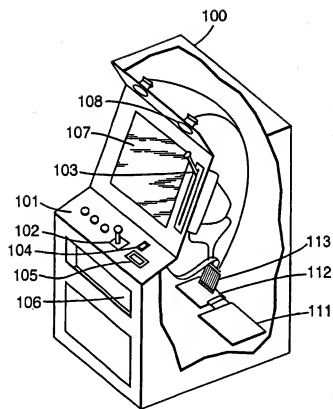of the universal gaming system.

82.   A gaming system suitable for use in a casino comprising:

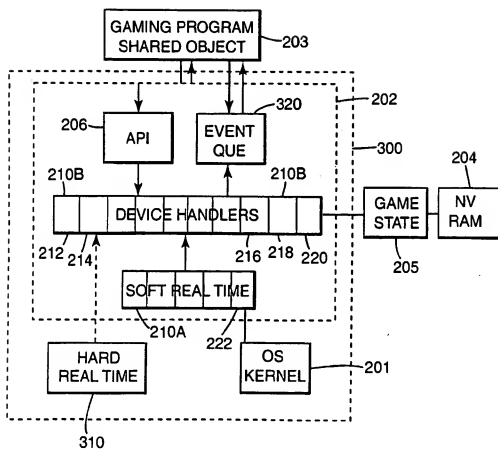   a game controller configured to operate the gaming system; and

       a first nonvolatile memory and a second nonvolatile memory for storing
critical gaming information, wherein the first nonvolatile memory and the second
nonvolatile memory are configured to communicate with the game controller as a gaming
RAID system for redundant storage of critical gaming information.

83.   The system of claim 82, wherein the first nonvolatile memory and the
second nonvolatile memory are random access memory.

Fig. 1

Fig. 2

Fig. 3
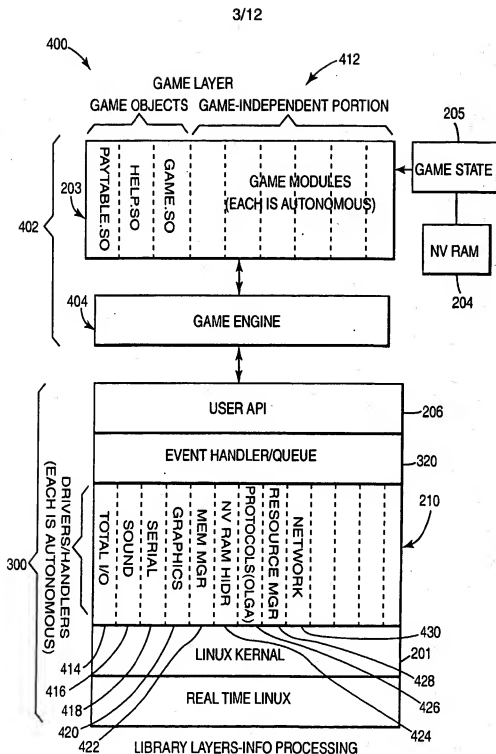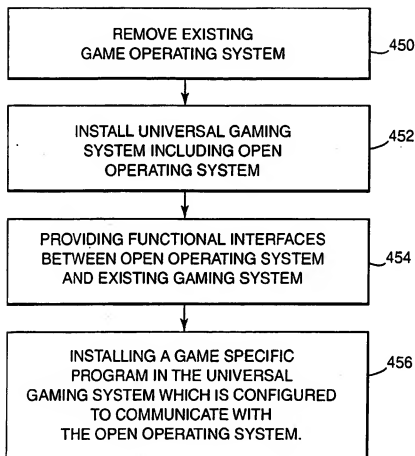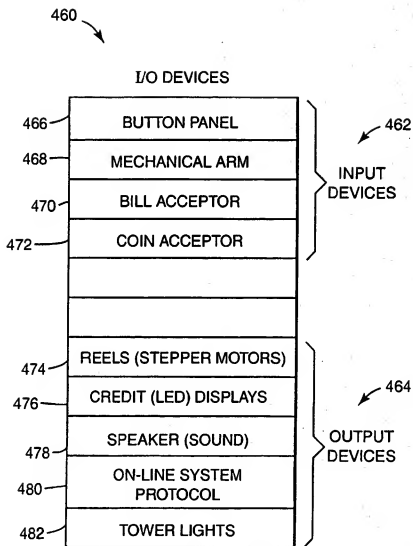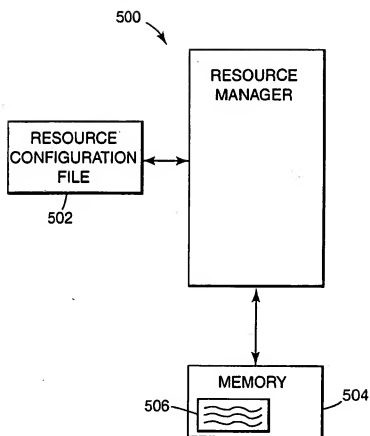
```
┌─────────────────────────────┐
│      REMOVE EXISTING         │ ╲
│   GAME OPERATING SYSTEM      │   450
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│   INSTALL UNIVERSAL GAMING   │
│   SYSTEM INCLUDING OPEN      │   452
│     OPERATING SYSTEM         │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ PROVIDING FUNCTIONAL INTERFACES │
│ BETWEEN OPEN OPERATING SYSTEM   │  454
│   AND EXISTING GAMING SYSTEM    │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│    INSTALLING A GAME SPECIFIC   │
│    PROGRAM IN THE UNIVERSAL     │  456
│ GAMING SYSTEM WHICH IS CONFIGURED │
│      TO COMMUNICATE WITH        │
│   THE OPEN OPERATING SYSTEM.    │
└─────────────────────────────┘
```

Fig. 4

460

I/O DEVICES

466 —	BUTTON PANEL
468 —	MECHANICAL ARM
470 —	BILL ACCEPTOR
472 —	COIN ACCEPTOR

462

INPUT DEVICES

474 —	REELS (STEPPER MOTORS)
476 —	CREDIT (LED) DISPLAYS
478 —	SPEAKER (SOUND)
480 —	ON-LINE SYSTEM PROTOCOL
482 —	TOWER LIGHTS

464

OUTPUT DEVICES

Fig. 5

Fig. 6

7/12

506

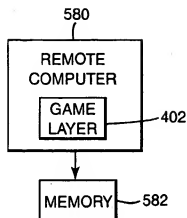|	
510 — PIN 1	R20 — 512
514 — PIN 2	R3 — 516
518 — PIN 3	R38 — 520
522 — PIN 4	R10 — 524
526 — PIN 5	R11 — 528
530 — PIN 6	R12 — 532
534 — PIN 7	R13 — 536
⋮	⋮
538 — PINN	R51 — 540

Fig. 7

Fig. 8

Fig. 9

Fig. 10

Fig. 11

620

WEB PAGE TEMPLATE

GAME TYPE —626

GAME SPECIFIC CONFIGURABLE OPTIONS

622—

628—

TEMPLATE 1

TEMPLATE 2

624
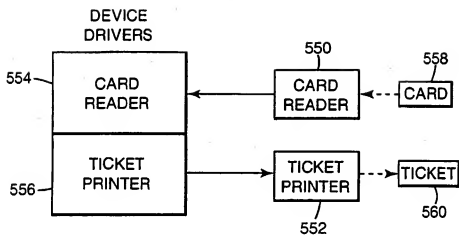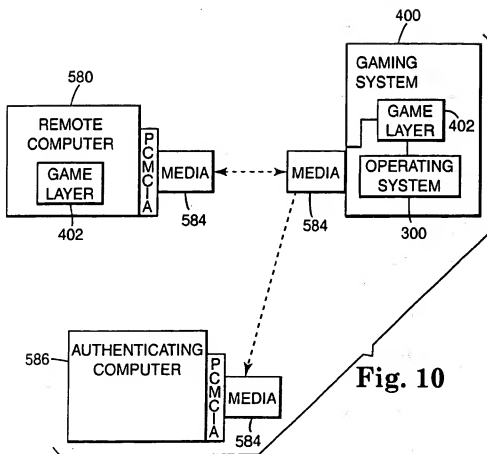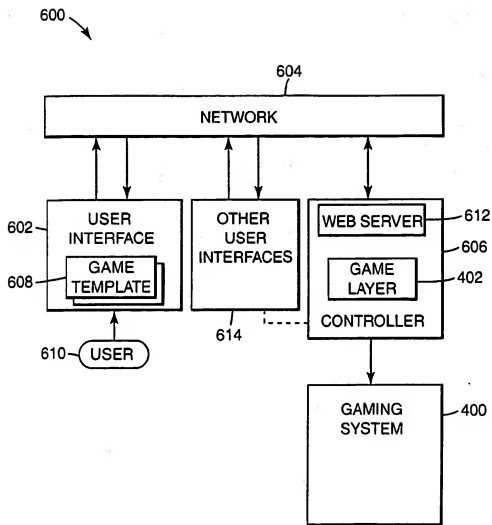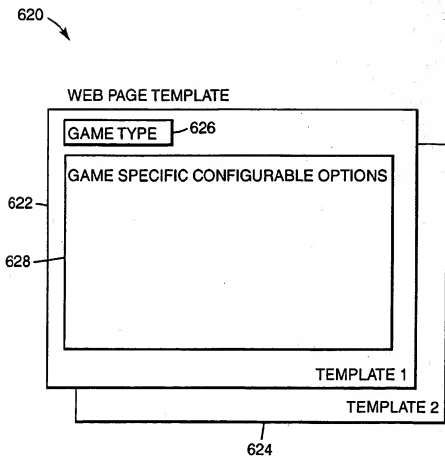
Fig. 12

Fig. 13

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : GO6F 19/00; GO6F 17/00
US CL : 463/16, 24, 28, 29; 709/1, 102, 328

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 463/16, 24, 28, 29; 709/1, 102, 328

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base, and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,889,990 A (COLEMAN et al) 30 March 1999 (230.02.1999); all	1-8, 10-16, 19-26, 28-45, 48-72, 75, 76, 79-81
Y	US 4,462,076 A (SMITH III) 24 July 1984, column 1, lines 6 - column 2, line 25.	9, 17, 44, 47, 78
Y	US 4,727,544 A (BRUNNER et al) 23 February 1988 (23.02.1978), column 1, line 12 - column 2, line 15.	9, 17, 44, 47, 78
Y	US 6,035,321 A (MAYS) 07 March 2000, (07.03.2000), column 1, line 7- column 5, line 50.	1-8, 10-16, 19-26, 28-44, 48-72, 75, 76, 79-81
Y	US 6,021,414 A (FULLER) 01 February 2000 (01.02.2000), abstract, column 1, line 23 - column 3, line 52.	1-8, 10-16, 19-26, 28-44, 48-72, 75, 76, 79-81, 18, 27, 44, 47, 78
A	US 6,026,238 A (BOND et al) 15 February 2000 (15.03.2000), column 1, 16 - column 2, line 42.	1-8, 10-16, 19-26, 28-44, 48-72, 75, 76, 79-81
A	US 6,006,279 A (HAYES) 21 December 1999 (21.12.1999), column 1, line 15 - column 2, line 67.	

☒ Further documents are listed in the continuation of Box C.   ☐ See patent family annex.

*	Special categories of cited documents:
"A"	document defining the general state of the art which is not considered to be of particular relevance
"E"	earlier application or patent published on or after the international filing date
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O"	document referring to an oral disclosure, use, exhibition or other means
"P"	document published prior to the international filing date but later than the priority date claimed

"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
08 May 2001 (08.05.2001)	12 JUN 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Steven Ashburn *Sheila Veney* *Paralegal Specialist* *Technology Center 3700* Telephone No. 703 308 1078

Form PCT/ISA/210 (second sheet) (July 1998)

C (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,003,038 A (CHEN) 14 December 1999 (14.12.1999), column 1, line 13 - column 2, line 67.	1-8, 10-16, 19-26, 28-44, 48-72, 75, 76, 79-81
Y	US 5,552,290 A (CALVERT et al) 03 September 1996 (03.09.1996), column 1, line 16 - column 3, line 27.	1-8, 10-16, 19-26, 28-44, 48-72, 75, 76, 79-81
Y	US 5,394,547 A (CORRENTI et al) 28 February 1995 (28.03.1995), column 1, line 7 - column 2, line 46.	1-8, 10-16, 19-26, 28-44, 48-72, 75, 76, 79-81
A	US 5,901,319 A (HIRST) 04 May 1999 (04.05.1999), column 1, line 7 - column 4, line 27.	1-8, 10-16, 19-26, 28-44, 8-72, 75, 76, 79-81
A	US 5,870,587 A (DANFORTH et al) 09 February 1999 (09.02.1999), column 1, line 10 - column 2, line 21.	1-8, 10-16, 19-26, 28-44